HUNTING AND GATHERING

Robert Filman • filman@computer.org

A few weeks ago I read a *New Yorker* article that asked various celebrities about their favorite *New Yorker* cartoons. (Don't ask me which issue; I read whatever *New Yorker* has been left near the StairMaster. After the StairMaster I'm in no condition to record bibliographic references.) In any case, through some quantum mechanical oversight, they missed asking moi. I would have told them my favorite was the one where the tie-and-jacketed gentleman places two bags of groceries on the kitchen counter and yells, "Hi Honey. I'm home from hunting and gathering."

I've always (or at least since I saw the cartoon on some other StairMaster evening) felt it was a wonderful expression of the Internet age, where mammals designed for fighting mastodons on the savanna have been reduced to fighting bugs in Windows 98. This issue's collection of sites was gathered in that spirit.

DISTRIBUTED RANDOMNESS
HotBits: Genuine Random Numbers • www.fourmilab.ch/hotbits/
John Walker

Too much order in your life? Visit John Walker's ant (fourmi) laboratory for some sequential chaos. Unlike mere pseudorandom number generators, fourmilab offers "genuine random numbers, generated by radioactive decay." (That is, genuinely random if you believe God plays dice with the universe.) For free. I asked for a random byte, and sure enough, the system replied (in hexadecimal) "11." Now if that's not a random byte, I'm not sure what is. HotBits also provides a Java package that can be used to programmatically obtain random sequences.

Walker gets the hunting and gathering award for bringing the full power of modern theoretical physics and communications technology to bear on the problem of creating lots of data with absolutely no information.

Walker was a founder of Auto-Desk, and is amusingly opinionated. While you're at the site, check out the satellite views of the earth (www.fourmilab.ch/earthview/). You can also read Walker's original science fiction stories (which are about as good as the average at Intertext (www.intertext.com) and a lot better than most of Quanta (www.quanta.org/quanta)); his history of AutoDesk (for which you have to have been there to care); and, of course, his diet book (which may not be a surer way to lose weight, but is at least cheaper than anything at Amazon).

AGENTS
Foundation for Intelligent Physical Agents • www.fipa.org/

I believe even Urk would have preferred getting someone else to chase the saber-tooth tigers for him. In contemporary terms, he needed an agent. So I turned to the Foundation for Intelligent Physical Agents (FIPA) to find out about agent standards.

FIPA wants to be a standards body for agent technology. If only we had standards for agents, progress would go much faster. And these standards

---

The Spider's rating system:

5 Flies = Scrumptious

4 Flies = Tasty

3 Flies = Filling

2 Flies = Edible

1 Fly = Yuck!

---

need to be promulgated by small, entrepreneurial organizations like FIPA, rather than large, clumsy dinosaurs like ISO or reality-detached academicians. On the Web site I found copious information about FIPA's organization, bylaws, and how to join. By looking really hard I even found various long documents on agent standards, documents more impressive by volume than weight.

Ah, but the "library" page offered the meat of the matter. There I found the following "strawman" definition of an agent: "An entity that resides in environments where it interprets 'sensor' data that reflect events in the environment and executes 'motor' commands that produce effects in the environment." The paragraph goes on to assure me that a hardware agent would thus include a lot of software. Possible, though not required, attributes of agents are specified as autonomy, social ability (interaction with other agents and/or humans), reactivity, pro-activeness (initiative for goal-directed behavior), mobility, temporal continuity, and adaptivity.

As near as I can tell, from this definition (McCarthy: http://www-formal.stanford.edu/jmc/ascribing.html), my thermostat is an agent (lacking only mobility and adaptivity). Moreover, if I'd sprung for the fancier HVAC, my car's thermostat would do it one better. (Keep in mind that my thermostat is not elaborate hardware/software integration, but a slice of twisted metal.)

The crux of FIPA's technology seems to be an agent communication language (ACL) which embellishes a KQML-ish syntax (www.cs.umbc.edu/kqml/) with more semantic intuitions, but without providing an underlying mechanistic interpretation or deeper semantic demands. It's got form without content. I hate to break the news to FIPA, but this is not a foundation for success.

I especially wanted to read the papers in the FIPA library that offered (1) an overview of FIPA, (2) an overview of the FIPA 97 standard, and (3) an assessment of FIPA 97, but FIPA neglected to connect the pull-down menu for these items to any actual links.

Acses Muenchhoff & Janz GmbH • www.acses.com/

After a long day of hunting and gathering, it's good to curl up by the campfire with a bargain book. It's better if you can get something else to gather that book for you. One of the more comprehensive shopping agents I've found is Acses, which searches twenty-five online bookstores for a requested title and sorts the results by cost, including shipping. Depending on what you're looking for, prices can be similar everywhere (as with recent or difficult-to-find books) or dramatically different (as with popular or formerly popular books).

I asked Acses to find David Flanagan's 1997 *Java in a Nutshell* (with CD-ROM), list price of $69.95. Acses reported domestic total costs ranging from $48.96 at Shopping.com (including 3–7 day UPS shipping) to $73.95 at CodysBooks.com. (I could also, for example, have had the book sent back to the USA by international express post from Dymocks, Australia, which would have cost $115.34.) On the other hand, if I wanted next-day delivery of the more current *Open Sources: The Cathedral and the Bazaar and Other Essays from the Leaders of the Open Source Movement*, edited by Mark Stone, and was bright enough not to try to get it here from England, there's little difference between Books.com at $30.86 and Amazon.com or BarnesandNoble.com at $30.91. (Of course, if I really wanted to read *The Cathedral and the Bazaar*, I'd just go directly to Eric Steven Raymond's page at www.tuxedo.org/~esr/writings/cathedral-bazaar/ where I could choose among HTML, SGML, PostScript, or ASCII, with no additional charge for same-day delivery.)

For difficult to find titles, like *Coordinated Computing: Tools and Techniques for Distributed Software*, prices ranged from $69.80 at Books.com (2–6 weeks) to $71.00 at BriansBooks.com (1–10 days)—an outrageously large sum for an out-of-date technical book (which may explain why I haven't seen any royalties lately). I am impressed, though, how quickly these various bookstores

promise to obtain an out-of-print book, or that they've even heard of one.

Moving from the technical to the recreational (after that long day of hunting and gathering, you didn't really want to curl up with *Coordinated Computing*, did you?), I sought Anne Tyler's *The Accidental Tourist*. I could get this mass-market paperback (list-priced at $7.50) in a week or two from seven sources, with prices that ranged from $6.75 at 1Bookstreet.com to $10.65 at WordsWorth.com. This search points out that large publicity budgets are no guarantee of low prices; BarnesandNoble.com had it for $9.54 and Amazon.com for $9.95. 1Bookstreet also has the virtue of differentiating between editorial and advertisements (www.nytimes.com/library/tech/99/02/biztech/articles/08amazon.html), although Amazon promises to do better in the future.

Acses rates a permanent spot on my bookmark list.

NOCTURNAL CONTEMPLATION OF BARBARA EDEN
Jini(TM) Technology • java.sun.com/jini/whitepapers/
Sun Microsystems

Raymond's *The Cathedral and the Bazaar* has become the manifesto of the open source movement, helping popularize the idea that in the best of all worlds, software would be built as a cooperative effort by volunteers. The product of such cooperation would then be made freely available. Surely one impetus to the open software movement is Microsoft's dominance in the personal computing office suite and operating systems' markets, and the potential for abuse that this dominance has enabled. (I find *Slate*'s (www.slate.com) coverage of this issue to be the most amusing, unpretentious, and remarkably free of bias for a Microsoft subsidiary, and wholeheartedly recommend the site, now that it's free again.)

The great NOISE (that is, the anti-Microsoft conspiracy: Netscape, Oracle, IBM, Sun, and everybody

else) hope has centered on open-source Linux (which has accumulated market penetration well into single digits) and on Sun's Java (a latest greatest thing for several years now). Java, originally presented as a universal programming language, is evolving toward being a universal computing environment. There is much virtue in this, as the traditional separation of operating systems and languages produces unnecessary cognitive discontinuities and convoluted programs.

Sun's latest step in the progression of Java to operating system is Jini. The Jini vision is that a "Jini device" is just plugged into a network. Then everyone on that network can access the device. Buy a new printer or disk drive, plug it in, and it works. Besides the operating system, Jini will let us dispense with system administration.

Jini implementation relies on three technologies already present in Java—distributed objects (Java RMI), interfaces, and code mobility; and two implementation tricks—traders and leases. On connecting to a network, a Jini device broadcasts a message seeking a trading agent. On finding one, it registers with that agent, indicating which interfaces it supports (from the universal interface space) and additional properties with respect to those interfaces. Thus, when you plug in a disk drive it could tell the trader that it does "disk drive," has a gigabyte of free space, and a transfer rate of 100 Mbytes/second. Plug in a printer, and it might announce its support of the "XYZZY.com color printing interface" (an extension of "color printing," itself an extension of "printing") and that its speed is "6 pages per minute."

A client application that seeks a color printer can inquire whether the trader knows of one (or perhaps even one that prints 6 pages per minute). The trader gives a reference to our friend above, and the client application can make calls on that printer. Since the printer looks like a Java object and uses Java datatypes, interfacing is straightforward. If the printer needs to run code on the client (that is, install a printer driver or display a GUI presenting printing choic-

es) then Java code mobility can seamlessly install it.

All promises of services are done with respect to "leases." That is, service is promised for a period of time. If the service contract is not renewed within that time, the connection goes away. This is a clever approach to distributed garbage collection, provided you're not trying to handle too many links or persistent objects.

The Sun Jini whitepapers will, if you read them closely enough, tell you all this. From the Spider's point of view, however, they suffer from two failings: beating around the bush to get to the point, and, more seriously, failing to discuss the limitations of the approach. The high chaff/wheat ratio seems endemic to almost everything on the Web labeled "whitepaper." More importantly, the whitepapers skim over the technical limitations of the above scenarios. Not only do operating systems provide connectivity to devices, they also provide protection and security. A (vanilla) disk drive has no idea who is reading or writing to it. Store something there, and anyone else who gets a reference to that drive (and the trader is advertising those references) can read it or change it. Or, as Sun phrases it, "Jini is for intranets" (since, presumably, intranets have no security concerns). A more complex

disk drive introduces the spectrum of system administration problems. The ability to extend interfaces with new interfaces produces a flourishing, Maoist-style, thousand blooming flowers. Wonderfully democratic, but I'd bet it's a long-term maintenance nightmare. (It said it was a color printer. How was I to know that it would bring up a dialog box in the middle of my batch job? You mean it "prints" to a screen display?)

The Jini trader model has its limitations, too. I may want a "fast enough" printer, but the trader can't be told that a 6 page-per-minute printer is a more than adequate substitute for a 4 ppm request. Similarly, I may not care where a disk drive is physically located, but I probably don't want my California hard copy appearing in Virginia. Thus, either the printer must be configured on plug-in (here come those pesky system administrators again), or, in addition to having a Java virtual machine, it's also going to need a GPS.

These complaints are not to say that Jini's not a good idea, or that it won't take over the world someday. I hope that happens, but I hope it's (1) after these technical kinks have been worked out and (2) after better technical writing about the system has appeared.

* * *

Arachnoid Tourist at *IC Online*

Check out *IC Online*'s archives for interviews, round tables, industry reports, and of course, the *Arachnoid Tourist Online* (http://computer.org/internet/arch.htm).

Seek, and Ye Shall Find. The Spider takes a tour of search engines and compares them by an absolutely precise, but utterly arbitrary, system (from Jul/Aug 1998).

Spider on the Go. The Tourist packs up his silk and explores mobile computing (from Mar/Apr 98).

Stuck in the Mud. The Tourist takes a look at virtual communities and their underlying technologies (from Jan/Feb 98).

An Enterprising Spider. Take a tour of electronic commerce sites with IEEE Internet Computing (from Nov/Dec 97).